
Lehrportfolio

Software-Architekturen und -Systeme

Autor: **Prof. Dr. René Wörzberger**

29.12.2018 / Version 2.0

Inhalt

1	Einleitung	3
2	Wer bin ich? - Mein Hintergrund	3
3	Was lehre ich? - Das Lehrgebiet	3
3.1	Was tut ein Software-Architekt?	3
3.2	Was sind die Herausforderungen im Software-Bereich?	3
3.2.1	Das Naturell der Software	3
3.2.2	Die Informatik als Disziplin	4
3.3	Meine Lehrveranstaltungen	5
3.3.1	Software-Management	5
	Learning-Outcome	5
	Lehrform	6
3.3.2	Systementwurfs-Praktikum	6
	Learning-Outcome	6
	Lehrform	7
3.3.3	Software-Architecture for Large Systems (Arbeitstitel).....	7
	Learning-Outcome	7
	Lehrform	8
3.3.4	Client-Server-Basics	8
	Learning-Outcome	8
	Lehrform	8
3.3.5	DevOps	8
	Learning-Outcome	8
	Lehrform	9
3.3.6	Software-Engineering.....	9
4	Warum und wozu lehre ich?	9
4.1	Meine Motivation	9
4.2	Ausgangslage und Motivation der Studierenden	9
4.3	Wichtige Inhalte des Aufgabengebiets.....	10
4.4	Anspruch an mich selbst und an Studierende	10
5	Wie lehre ich?	11
5.1	Orientierung durch Struktur	11
5.2	Aktivierende Lehre über Clicker-Systeme.....	11
5.3	Hands-On-Übungen an eingesetzten Tools.....	11
6	Mit welcher Wirkung lehre ich?	11
7	Wohin soll es gehen?	12
7.1	Reduktion der frontalen Lehre.....	12
7.2	eLearning	12
7.3	Didaktische Methoden.....	12
7.4	Wissensbeschaffung lernen	12
	Anhang	13

1 Einleitung

Dieses Dokument ist im Rahmen des einjährigen LehrendenCoaching-Programms für neuberufenen Professoren in der TH Köln entstanden. Es stellt mein Lehrangebot, angewendete Lehrmethoden und meine Lehrauffassung dar.

2 Wer bin ich? - Mein Hintergrund

Ich bin seit März 2018 als Professor für Informatik, insbesondere "Software-Architekturen und -Systeme" an der Fakultät 7 - Informations-, Medien- und Elektrotechnik an der TH Köln tätig.

Zuvor war ich ein Semester als Professor für Informatik, insbesondere "Software-Engineering" an der Hochschule Düsseldorf beschäftigt.

Von 2010 bis 2017 war ich in verschiedenen Positionen in der Industrie tätig: als technischer Projektleiter und Berater bei NTT DATA Deutschland und als IT-Architekt und Projektleiter bei der Deutschen Post DHL GmbH. Mein Schwerpunkt war hierbei die Architektur datenintensiver, webbasierte Systeme. In dieser Zeit habe ich viel über die "real existierende Informatik" gelernt, die ich in die Lehre einfließen lasse.

Davor habe ich meine formelle Ausbildung mit einer Promotion im Bereich der Prozessmanagement-Systeme am Lehrstuhl für Softwaretechnik (Software-Engineering) an der RWTH Aachen abgeschlossen. An der RWTH Aachen habe ich zuvor auch Informatik studiert.

3 Was lehre ich? - Das Lehrgebiet

Die Informatik selbst ist im Kern eine mathematisch-naturwissenschaftliche als auch ingenieurwissenschaftliche Disziplin mit weiteren Anleihen aus weiteren Disziplinen wie z.B. dem Design.

Ein Teildisziplin der Informatik ist das Software-Engineering, das sich mit der ingenieurmäßigen Entwicklung von Software beschäftigt. Zum Software-Engineering gehört insbesondere das Gebiet der "Software-Architekturen und -Systeme", was meine Denomination ist.

3.1 Was tut ein Software-Architekt?

Der Begriff der Software-Architektur ist dem Bauwesen entlehnt: Architekten entwerfen Produkte (Gebäude bzw. Software), wobei Sie verschiedene, einander teils widerstrebende Interessen und Ziele berücksichtigen müssen. Sie beschäftigen sich aber weniger mit Projektleitungsaufgaben, dem tatsächlichen Bau des Produkts (programmieren bzw. mauern, betonieren etc.) oder Betrieb, auch wenn ein tiefgehendes Wissen auch in diesen Gebieten teils unerlässlich ist.

3.2 Was sind die Herausforderungen im Software-Bereich?

3.2.1 Das Naturell der Software

Ein scherzhafte Definition des Begriffs "Software" lautet: "Software ist das, was nicht kaputtgeht, wenn's runterfällt.". Software ist schließlich ein immaterielles Produkt. Als solches hat die Entwicklung eines Software-Produkts im Vergleich zu anderen Ingenieursprodukten (Gebäude, Motoren etc.) Vor- und Nachteile:

- Software kann leicht (re-)produziert und verteilt werden. Während Monate zwischen Bestellung und Lieferung eines Neuwagens vergehen, ist eine neue App auf dem Smartphone nur wenige Klicks entfernt.

- Software ist leicht änderbar. Weitere Funktionen sind in Software schnell implementiert, wo bei anderen Ingenieursprodukten Produktionsstraßen aufwändig umgebaut werden müssten oder das Produkt von Grund auf neu entwickelt werden muss. Diese Eigenschaft ist für die Software gleichzeitig Fluch und Segen. Das Wissen, dass Software gut änderbar ist, führt oft zu einer unsystematischen Erweiterung. Dem entgegenzuwirken ist eine Aufgabe von Software-Architekten.
- Software ist schwer modellierbar. Im Bauwesen ist das maßstabsgerechte Modell ein guter Ausgangspunkt, ein gemeinsames Bild in den Köpfen der Interessengruppen (Bauherren, Nutzer, Statiker etc.) zu schaffen. Viele wesentliche Eigenschaften lassen sich aus diesem einen Modell ableiten. Die geeignete Modellierung und Dokumentation von Software hingegen gelingt nur über verschiedene, auf bestimmte Interessengruppen zugeschnittene Sichten.

3.2.2 Die Informatik als Disziplin

Der Megatrend der "Digitalisierung" ist in aller Munde. Auch in klassischen Industrien findet sich Software allerorten, in der automatisierten Wertschöpfungskette (Stichwort: automatisierte Produktionsstraßen), im Produkt selbst (Beispiel: Car-Entertainment-Systeme) bis hin neuen Marketing- und Vertriebskanälen (Stichwort: Social Media). Dementsprechend gut sind die Berufsaussichten für Informatiker.

Die Informatik ist ein dynamisches Feld. Es gibt viele Ansätze und Ansichten, wie Software zu entwickeln ist. Scherzhaft wird in manchen Bereichen von einer "kambrischen Explosion" verfügbarer Hilfsmittel gesprochen, darunter verschiedene Programmiersprachen, -frameworks und -tools. Daher führen hier viele Wege nach Rom; der Grad der Standardisierung und Normierung ist dementsprechend niedrig. Häufig durchlaufen Architektur-Trends wie "Service-orientierte Architekturen" Phasen überzogener Erwartungen. Sie "oszillieren" zwischen Extremen und kehren in neuer Gestalt wieder (Beispiel: "Vom Host-Terminal (70er Jahre) zur Desktop-Applikation (80er/90er) zum Web-1.0-Browser (90er/00er) zu Apps und Web-2.0-Browsern (10er)"). Sie begünstigen einander (Stichwort: "Cloud, Container-Virtualisierung, Agilität und Microservices") und fristen so manchmal vor ihrem Hype ein jahrzehntelanges Nischendasein (Stichwort: "Künstliche Intelligenz"). Methoden werden aus anderen Disziplinen übernommen (Beispiel: tayloristische Prozesssteuerung von Arbeitsabläufen) und wieder durch dem Software-Naturell angemessenere ersetzt (Stichwort: Agilität).

Die Informatik ist noch junges Feld in zweierlei Hinsicht: Zum einen wurden die ersten Computer erst Mitte des vergangenen Jahrhunderts gebaut, die Disziplin des Software-Engineerings erst Ende der 60er-Jahre aus der Taufe gehoben. Mathematiker, Physiker und Ingenieure haben davor in ihren Disziplinen schon Jahrhunderte bis Jahrtausende gewirkt. Da in der Informatik sind nur einige formale Grundlagen beweisbar und unwiderlegbar richtig sind, besteht ein Großteil des Informatikwissens aus Erfahrungswissen, das aufgrund der Juniorität die zuvor beschriebene Dynamik aufweist und im stetigen Fluss ist. Zweitens wächst die Zahl der Beschäftigten im IT-Bereich und damit zwangsweise der Junioren mit wenig Berufserfahrung stark¹.

Diese Eigenschaften von Software und der Informatik als Disziplin bilden wesentliche Rahmenbedingungen für die Inhalte meiner Lehre.

¹ Laut statista.com hat sich die Zahl der IT-Beschäftigten in Deutschland zwischen 2007 und 2018 von ca. 550.000 auf ca. 950.000 nahezu verdoppelt.

3.3 Meine Lehrveranstaltungen

In verschiedenen Studiengängen führe ich folgende Lehrveranstaltungen durch bzw. habe oder werde ich durchführen:

- Studiengang "Bachelor Technische Informatik (BaTIN)" an der TH Köln am Standort Deutz
 - Lehrveranstaltung "Software-Management"
 - Lehrveranstaltung "Softwareentwurfs-Praktikum"
- Studiengang "Master Technische Informatik (MaTIN)" an der TH Köln am Standort Deutz
 - Lehrveranstaltung "Software Architecture for Large Systems (Arbeitstitel)"
- Studiengang "Code & Context" an der TH Köln am Standort Mülheim (in Entwicklung)
 - Lehrveranstaltung "Client-Server-Basics (Arbeitstitel)"
 - Lehrveranstaltung "DevOps (Arbeitstitel)"
- Studiengänge "Bachelor/Master Medieninformatik" an der Hochschule Düsseldorf
 - Lehrveranstaltung "Software-Engineering"

3.3.1 Software-Management

Wenn eine neuentwickelte Software(-Version) auf dem Rechner eines Entwicklers funktioniert, ist erst der halbe Weg zur eigentlichen Wertschöpfung zurückgelegt. Die zweite Hälfte des Wegs ist, die Software zusammen parallel entwickelten Versionen qualitätsgesichert und möglichst automatisiert in durchaus komplizierte Betriebsumgebungen zu überführen und dort zu betreiben.

Die Vorlesung "Software-Management" befasst sich daher weniger mit dem Entwurf und der Konstruktion von Software, sozusagen den "Innereien" der Software, sondern mit dem "Drumherum". Studierenden werden befähigt Versionsverwaltungssysteme, Build-Systeme und Werkzeuge zur Automatisierung der Qualitätssicherung anzuwenden. Diese machen die arbeitsteilige Entwicklung großer Software-Systeme erst praktikabel. Des Weiteren können Studierende durch diese Lehrveranstaltung Software in moderne Cloud-Betriebsumgebungen installieren und betreiben.

Learning-Outcome

Das Learning-Outcome der Lehrveranstaltung ist daher folgendermaßen formuliert:

(WAS) In systematischer Weise versionieren und verteilen Studierende den Quellcode eines gemeinsam entwickelten Systems im Team, automatisieren Maßnahmen zur Qualitätssicherung und zur Überführung in ein lauffähiges System und betreiben das System dem Stand der Technik entsprechend,

(WOMIT) indem sie Prinzipien der Quellcode-Versionverwaltung anhand des Tools "Git" praktisch anwenden, gängige Tools zur Software-Build-Automatisierung und Automatisierung der Qualitätssicherung anwenden und deren Ausgaben korrekt bewerten können und eine Cloud-Umgebung (z.B. Google Cloud, Amazon AWS oder Microsoft Azure) zweckmäßig einrichten und das System darin installieren, überwachen und kontinuierlich aktualisieren,

(WOZU) damit Sie später das moderne Berufsbild eines Software-Engineers besser ausfüllen können, das über das des reinen Programmierers hinausgeht.

Lehrform

Die Form der Lehre ist eine Mischung aus Vorlesung mit integrierter Übung (3 SWS) sowie einem Praktikum (2 SWS). Die Übungen sind insoweit in die Vorlesung integriert, als sie im ständigen Wechsel mit ihr stattfinden. Eine typische Lehrinheit (Vorlesungs- und Übungstermin) besteht aus drei Themen mit jeweils ca. 30 Minuten Vorlesung und direkt anschließender Übung von 20 Minuten.

Das Praktikum ist letztlich eine geführte Projektaufgabe, die die Studierenden im Laufe des Semesters als Hausaufgabe bearbeiten müssen. In Teams mit je zwei Mitgliedern wenden Sie auf ein beispielhaftes Software-System erlernte Methoden an. Sie werden dabei durch eine recht präzise gefasste Aufgabenstellung mit zahlreichen Unteraufgaben geführt. Ergebnisse und Zwischenergebnisse werden aus Sicht der Studierenden an drei Praktikums-Präsenzterminen im Semester überprüft und besprochen. Zur Überprüfung und Sicherstellung des Lernerfolgs beider Teammitglieder wird in den Präsenzterminen auch eine Anwesenheitsaufgabe mit ca. 30-minütiger Bearbeitungsdauer gestellt.

Die Lehrveranstaltung "Software-Management" ist als Wahlmodul dem Studiengang "Bachelor Technische Informatik" zugeordnet.

3.3.2 Systementwurfs-Praktikum

Im Curriculum des Studiengangs "Bachelor Technische Informatik" werden die Studierenden stufenweise an die selbständige, teambasierte Bearbeitung einer Software-Entwicklungsaufgabe herangeführt. Basierend auf den Programmiergrundveranstaltungen "Praktische Informatik 1 und 2" absolvieren die Studierenden ein Programmier-Praktikum, in dem sie einzeln eng gefasste und kleine Programmieraufgaben lösen. Im vierten Semester lernen Sie im Software-Praktikum, wie Sie auf Basis ausformulierte Spezifikationen im Team von ca. vier Teilnehmern ein größeres Software-System implementieren, d.h. im Detail ausspezifizieren und kodieren ("ausprogrammieren"). Im fünften Semester wird der letzte Schritt hin zu einem realistischen Software-Entwicklungsprojekt vollzogen. Im Systementwurfs-Praktikum werden die Studierenden in ca. 20 verschiedenen Projekte eingeteilt zu je zwei bis fünf Teilnehmern. Die Projektthemen speisen sich i.d.R. aus aktuellen Forschungsprojekten und -kooperationen der Lehrenden im Studiengang. Die Studierenden bekommen hier keine ausformulierten Spezifikationen, sondern müssen diese selbst erarbeiten. Zwischen- und Endprodukte werden in zwei Präsentationsterminen vorgestellt und am Ende des Semesters wird das fertige Software-Produkt inkl. geforderter Dokumentationen dem jeweiligen Projektbetreuer übergeben.

Learning-Outcome

Das Learning-Outcome der Lehrveranstaltung lässt sich folgendermaßen formulieren:

(WAS) Die Studierenden durchlaufen in der Rolle eines Auftragnehmers sämtliche Phasen eines Software-Entwicklungsprojekts, d.h. sie realisieren ein einsatzfähiges System auf Basis teils vager Auftraggeberanforderungen,

(WOMIT) indem Sie Anforderungen systematisch erheben und konkretisieren, Aufgaben im Team aufteilen und eine Zeitplanung erstellen, das System zunächst nur entwerfen bevor sie einen ersten Prototypen implementieren, den sie gegenüber ihrem Auftraggeber präsentieren und schließlich zu einem einsatzfähigen System ausbauen,

(WOZU) um später als Software-Engineer erfolgreich im Team Software-Entwicklungsprojekte im gesamten Projektzyklus bearbeiten zu können.

Lehrform

Als Verantwortlicher der Lehrveranstaltung übernehme ich querschnittliche und übergreifende Aufgaben, wie beispielweise die Themeneinwerbung, Zuteilung und die Organisation von Meilensteinsitzungen inkl. der Präsentationstermine aber auch die Bewertung der Arbeitsergebnisse hinsichtlich bestimmter Aspekte. Zudem Sorge ich in ausgewählten Aspekten für eine projektübergreifende Standardisierung der Entwicklungsprozesse in den Projekten. Beispielsweise werden alle Teilnehmer auf einen bestimmten technischen Prozess zur Lieferung von Ergebnissen verpflichtet. Außerdem erhalten die Studierenden einen Satz von Dokumentvorlagen und ein Vorgaben-Dokument. Diese Standardisierung erleichtert die Arbeit aus Sicht des Lehrenden und ist gleichzeitig ein wichtiges Lernziel: Derartige Standards sind auch im Berufsleben bei mittelgroßen und großen Arbeitgebern mit vielen parallelen Entwicklungsprojekten gängige Praxis. Ein Projekterfolg hängt nicht zuletzt auch an deren Einhaltung.

Eine Besonderheit dieser Lehrveranstaltung ist, dass sie mit der Lehrveranstaltung "Präsentation und Kommunikation (PuK)" durchgeführt wird. Studierende lernen in Seminaren der Kompetenzwerkstatt der TH Köln Grundlagen der Präsentation und Kommunikation.

Die Lehrveranstaltung "Softwareentwurfs-Praktikum" ist als Pflichtveranstaltung dem Studiengang "Bachelor Technische Informatik" zugeordnet.

3.3.3 Software-Architecture for Large Systems (Arbeitstitel)

Im Sommersemester 2019 werde ich erstmalig eine Lehrveranstaltung mit dem Arbeitstitel "Software-Architecture for Large Systems" anbieten.

Studierende werden hier mit Herausforderungen konfrontiert, die erst bei großen, datenintensiven Software-Systemen relevant sind, beispielsweise dem Spannungsfeld von Performance und Verfügbarkeit von Systemen bei gleichzeitiger Garantie von Datenkonsistenz. Zudem wird der Tätigkeitsbereich und das Handwerkszeug des IT-Architekten näher betrachtet: Studierende müssen in der Rolle Interessen sämtlicher Interessengruppen in einem komplexen Entwicklungsprozess kennen und miteinander vereinbaren können. Sie sollen zudem wissen, welche ehernen Prinzipien dem Entwurf eines Systems zugrunde liegen und mit welchen Modellen, Sichten und Dokumentationsformen Architekturen zielgruppengerecht darstellen lassen.

Learning-Outcome

Das Learning-Outcome der Lehrveranstaltung ist folgendermaßen formuliert:

(WAS) Studierende können IT-Architekturen für verteilte, komplexe, geschäftskritische und in Enterprise-Systemlandschaften integrierte Software-Systeme entwerfen, implementieren und die Systeme in der Cloud betreiben,

(WOMIT) indem Sie

- verschiedene Stakeholder-Gruppen kennen und deren mitunter konfligierende Interessen abwägen,
- ebenso Qualitätskriterien (z.B. Security, Zuverlässigkeit, Skalierbarkeit und Wartbarkeit) und deren widerstrebende Kräfte gegeneinander abwägen,
- die Wirkung auf Qualitätskriterien und Stakeholder-Interessen (z.B. Informationssicherheit, Betriebskosten) in Entwurfsentscheidungen und im Einsatz von Architektur-Mustern und -Stilen (z.B. Entkopplung durch Messaging/Streaming, stärkere Verteilung durch Microservices, Betrieb in der Cloud, Verwendung georeduzierter Datenreplikation etc.) einbeziehen,
- dabei Trends (z.B. Microservice-Architekturen) auch kritisch hinterfragen,

- architekturelevante Anforderungen (z.B. niedrige Response-Zeiten) von irrelevanten Anforderungen (z.B. an das UX-Design) trennen und
- auf Basis von Szenarien schärfen,
- Entwürfe mit geeigneten Sichten, Notationen und Werkzeugen Stakeholder-spezifisch darstellen,
- zur Implementierung von Entwurfsentscheidungen geeignete Methoden (z.B. RESTful API-Design) und Werkzeuge (z.B. API-Spezifikation mit OpenAPI in Microservice-Architekturen) einsetzen sowie
- zum Betrieb realisierter Systeme in der Cloud verfügbare Dienste (z.B. verschiedene Storage-Arten) und Werkzeuge (z.B. Docker und Kubernetes) geeignet auswählen und anwenden,

(WOZU) um

- in Folgeveranstaltungen / Abschlussarbeiten (für externe oder interne Auftraggeber) langfristig verwendbare Prototypen realisieren zu können und um
- im Berufsleben in der Rolle als IT-Architekt im Enterprise-Umfeld arbeiten zu können.

Lehrform

Die Lehrform wird auch hier die Vorlesung mit integrierter Übung und parallelem Praktikum sein. Sofern sinnvoll, wird die Lehrveranstaltung auf Englisch angeboten.

3.3.4 Client-Server-Basics

Im Studiengang "Code & Context" wird der Kurs "Client-Server-Basics" im ersten Semester angeboten. Der Start des Studiengangs ist voraussichtlich das Wintersemester 2019/20.

Learning-Outcome

Die Studierenden können in Einzelarbeit auf Basis exakter und vollständiger Anforderungen eine webbasierte Client-Server-Applikation codieren, indem sie einen gegebenen Entwicklungs-Stack einsetzen, um später bei der Codierung großer derartiger Applikationen mitwirken zu können sowie alternative Frameworks und Libraries einordnen zu können.

Inhaltlich werden Browser-, Application-Server und Datenbank-Technologien gelehrt.

Lehrform

Der Kurs findet in einem zweiwöchigen Block statt. Der Tagesablauf besteht sowohl aus Impulsvorlesungen und praktischen Übungen.

3.3.5 DevOps

Ebenfalls im Studiengang "Code & Context" wird im dritten Semester die Lehrveranstaltung "DevOps" angeboten. DevOps ist ein Akronym bestehend aus Development (Entwicklung) und Operations (Betrieb).

Learning-Outcome

Die Studierenden können fortwährend entwickelte Software-Systeme automatisiert bauen und automatisiert in von ihnen aufgesetzten virtualisierten Cloud-Umgebungen installieren, indem sie

- Grundkenntnisse in Linux (als dem marktbeherrschenden Server-Betriebssystem) besitzen und zur Administration einsetzen können,
- Build, Test und Deployment der Software-Systeme über eine CI/CD-Pipeline automatisieren,
- grundlegende Administrationskenntnisse (Firewall-, Proxy-, Domain-Konfiguration, ...) nutzen,
- und Werkzeuge zur Virtualisierung und zum Cloud-Computing anwenden,

damit sie digitale Produkte agil über den gesamten Lebenszyklus und im Sinne einer Ende-zu-Ende-Verantwortung ("you build it - you run it") realisieren können.

Inhaltlich ist die Lehrveranstaltung stark verwandt mit Software-Management (s. Abschnitt 3.3.1)

Lehrform

Der Kurs findet in einem zweiwöchigen Block statt. Der Tagesablauf besteht sowohl aus Impulsvorlesungen und praktischen Übungen.

3.3.6 Software-Engineering

An der Hochschule Düsseldorf habe ich die Lehrveranstaltung Software-Engineering in den Studiengängen Medieninformatik Bachelor und Medieninformatik Master angeboten.

Die Studierenden wurden hier in das weite Feld des Software-Engineerings eingeführt.

Die Lehrform war die einer Vorlesung und eines begleitenden Praktikums.

Die Lehrveranstaltung wird derzeit nicht von mir angeboten.

Die Lehrveranstaltung wurde mit einer Gesamtnote im Zweierbereich evaluiert. Im Detail hat die Evaluation ergeben, dass Studierende den Praxisbezug sehr zu schätzen wussten und in gleichzeitig stattfindenden Projekten anwenden konnten. Beklagt wurde der permanent hohe Aufwand, der bei der Bearbeitung der Praktikumsaufgaben entstand.

4 Warum und wozu lehre ich?

4.1 Meine Motivation

Die langjährige Beschäftigung mit meinem Fach hat nie zu einer Sättigung des Wissenshungers geführt. Im Gegenteil interessiere ich mich für immer mehr Aspekte und neue Entwicklungen. In der industriellen Praxis bleibt leider zu oft zu wenig Zeit für eine angemessene Vertiefung eines Themas oder die angemessene Reflektion der bisherigen Vorgehensweise. Die Hochschule bietet hier bessere Möglichkeiten.

Die Lehre biete hierbei besondere Möglichkeiten: Wer lehrt, lernt. Das bedeutet, dass die Aufbereitung eines Themas für die eigene Lehre eine tiefgreifende Auseinandersetzung und auch Detailwissen voraussetzt: Man versteht die Dinge und Zusammenhänge genau.

Darüber hinaus motiviert mich, dass mein Wissen und an manchen Stellen auch meine subjektive Lehrmeinung an Studierende weitergeben kann. Umgekehrt lerne ich vor allem durch Projektarbeiten und Individualarbeiten selbst kontinuierlich hinzu.

4.2 Ausgangslage und Motivation der Studierenden

Die Lehre der Software-Architektur, wie auch des Software-Engineerings, steht vor der besonderen didaktischen Herausforderung, dass viele Studierende noch niemals an der Entwicklung größerer

Software-Systeme bzw. in größeren Software-Entwicklungsprojekten mitgewirkt haben. Ein ausgeprägtes Bewusstsein der Probleme, mit denen sich diese Gebiete befassen, kann nicht vorausgesetzt werden. Vielmehr erscheinen die Inhalte dieser Lehrgebiete den meisten Studierenden als unnötig akademisch und bürokratisch.

Entsprechend wichtig ist es, die entsprechenden Probleme größerer Entwicklungsprojekte plastisch und fallbezogen darzustellen: Nichts ist der Motivation der Studierenden zuträglicher als die Aha-Erlebnisse, in denen eine architektonische Abstraktion tatsächlich praktisch funktioniert. Ein Beispiel hierfür wäre der funktionierende Austausch einer gekapselten Modulimplementierung durch eine andere. Folglich ist die Wissensvermittlung über den vorlesungsbegleitenden Praktikums- und Übungsbetrieb mindestens so wichtig, wie die Vorlesung selbst.

4.3 Wichtige Inhalte des Aufgabengebiets

Das Gebiet der Software-Architekturen besitzt noch keinen allgemeinen akzeptierten Lehr-Kanon. Meiner Überzeugung nach stechen jedoch folgende, wichtige Themen heraus und sollten Studierenden in jedem Fall vermittelt werden:

Architektur-Prinzipien: Eherne Prinzipien wie das der „loosen Kopplung“ treten in der täglichen Architektur-Arbeit an vielen Stellen zutage. Ihre Einhaltung wirkt sich immer positiv auf eine Architektur aus, egal auf welchen Ebenen.

Architektur-Kommunikation: Die Rolle des Architekten umfasst insbesondere die Darstellung von komplexer Sachverhalte gegenüber verschiedenen Stakeholdern. Hier ist es besonders wichtig zu vermitteln, dass neben der situationsabhängigen Wahl der jeweils passenden Darstellungsform (Diagramm-Art etc.) auch das Diskussions-Niveau (beispielhaft vs. formal präzise) wichtig ist.

Architektur-Einflussfaktoren: IT-Architekturen existieren auf verschiedenen Maßstabsebenen: Kleine Software-Systeme besitzen ebenso eine Architektur wie große IT-Landschaften bestehend aus hunderten Systemen. Auf der Ebene großer IT-Landschaften wirken Einflüsse bei architektonischen Entscheidungen, die auf Ebene kleiner Software-Systeme keine Rolle spielen. Beispiel hierfür wären Kommunikationshemmnisse, Release-Zyklen oder kaufmännische Abschreibungsanforderungen. Studierende sollen nach Möglichkeit alle relevanten Einflussfaktoren kennenlernen, die bei der Gestaltung von Architekturen berücksichtigt werden müssen.

Architektur-Muster: Neben abstrakten Architektur-Prinzipien bilden sich in Software-Architekturen wiederkehrend gleiche Muster heraus. Studierende sollten die wichtigsten Muster kennen und anwenden können.

4.4 Anspruch an mich selbst und an Studierende

Ich stelle an mich selbst den Anspruch, Lehrinhalte interessant zu gestalten und didaktisch geeignet zu vermitteln. Dabei glaube ich, dass im Allgemeinen das *Abstraktionsvermögen* von Studierenden von einer konkreten Situation (Fallbeispiel) hin zu einer allgemeinen Gesetzmäßigkeit (Theorie) unterschätzt wird. Umgekehrt wird das *Interpretationsvermögen* von abstrakten Theorien hin zu konkreten Situationen jedoch überschätzt. Es fällt Studierenden leichter, hinter konkreten Fallbeispielen allgemeine Gesetzmäßigkeiten zu erkennen, als sich zu allgemeinen, abstrakt vermittelte Gesetzmäßigkeiten, konkrete Problemsituationen vorstellen zu können. Folglich muss Lehre immer induktiv und beispielhaft sein. Diese so zu gestalten ist ein Anspruch an mich, der tatsächliche Vollzug von Abstraktionen umgekehrt ein Anspruch an Studierende.

Ob dieses Ziel erreicht wird, lässt sich durch Transferaufgaben überprüfen. Hat ein Studierender beispielsweise das Prinzip der losen Kopplung auf „Java-Ebene“ verstanden, kann er davon abstrahieren und es beispielsweise auf Architekturen im Großen oder auch auf den Bereich des Projektmanagements (Aufgabenteilung) anwenden.

5 Wie lehre ich?

Um die in Kapitel 4 dargestellten Learning-Outcomes zu erreichen, verwende ich verschiedene Mittel in meinen Lehrveranstaltungen, die im Folgenden exemplarisch beschrieben werden.

5.1 Orientierung durch Struktur

Ich gliedere meine Lehrveranstaltungen in Lerneinheiten, wobei eine Lerneinheit einer Vorlesung inkl. Übung zzgl. Praktikum, also einer Semesterwoche entspricht. Eine Vorlesung inkl. Übung beginne ich i.d.R. mit der Rekapitulation des bisher Erlernten und der Einordnung der nun folgenden Lerneinheit. Dabei beziehe ich die Studierenden aktiv durch Fragen an das Auditorium ein.

Es folgt dann die Präsentation des Zeitplans der aktuellen Lerneinheit, d.h. welche Inhalte und zugehörigen Übungen in welcher Reihenfolge und mit welchem zeitlichen Aufwand behandelt werden. (Hier hat mir das Lehrenden-Coaching sehr geholfen: Studierende verlangen viel stärker nach Struktur einer Lehrveranstaltung, als mir das bislang bewusst war.)

5.2 Aktivierende Lehre über Clicker-Systeme

In der Lehrveranstaltung "Software-Engineering" habe ich "ARSNova" eingesetzt, ein sogenanntes Audience-Response-System. Über ARSNova wurden Studierende in der Vorlesung angeregt, eingestreute Fragen -- zumeist in Form von Multiple-Choice-Fragen -- anonym zu beantworten. Die Anonymität sichert hierbei eine höhere Beteiligung als bei direkt ans Auditorium gerichtete Fragen, die mangels Anonymität aus Sicht der Studierenden immer das Risiko der Blamage bergen.

5.3 Hands-On-Übungen an eingesetzten Tools

In der Lehrveranstaltung "Software-Management" wechsele ich zwischen Impulsvorträgen und Übungen ab. In den Übungen rege ich Studierende an, Aufgaben, die i.d.R. die Form eines Tutorials zu eingesetzten Tools mit einfachen, eingestreuten Fragestellungen haben, zunächst in einem Zeitfenster von 5 bis 10 Minuten selbständig zu lösen. Danach führe ich die Lösung selbst vor und gehe dabei auf zuvor entstandene Rückfragen der Studierenden ein.

Die direkte und aktive Beschäftigung mit konkreten Tools ist für mich Ausdruck meiner in Abschnitt 4.4 dargestellten Lehrauffassung. Ich bin überzeugt, dass Studierende eherne Prinzipien und Muster selbständig erkennen oder zumindest besser verinnerlichen können, wenn sie sie in praktischen Übungen selbst erfahren.

6 Mit welcher Wirkung lehre ich?

Bislang wurden zwei meiner Veranstaltung im EvaSys-Verfahren evaluiert. Die Ergebnisse befinden sich im Anhang. In Schulnoten ausgedrückt, wurden meine Lehrveranstaltung im Bereich einer 2 bewertet. Es freut mich, dass die Relevanz meiner Lehrinhalte von den Studierenden anerkannt wird. Ein wiederkehrender Kritikpunkt sticht hier heraus: Die Studierenden können sich kein Bild von der abschließenden Prüfung (Klausur) machen. Kurzfristig helfe ich hier gegen Ende der Vorlesung durch klausurnahe Übungsaufgaben ab. Langfristig möchte ich i.S.d. Constructive-Alignments bereits bei Beginn der Lehrveranstaltung ein genaueres Bild von Prüfungsinhalten vermitteln. Dies ist

aus meiner Sicht jedoch kein einfaches und risikoloses Unterfangen: In jedem Fall vermeiden möchte ich, dass Studierende nur für die abschließende Prüfung lernen und Lehrinhalte danach selektiv aufnehmen.

Der berechtigten Kritik an meiner verbesserungsfähigen Rhetorik begegne ich durch die Teilnahme an entsprechenden Workshops (s. Anhang)

7 Wohin soll es gehen?

7.1 Reduktion der frontalen Lehre

Die Lehrform der Vorlesung ist überspitzt formuliert ein Anachronismus. Vor Erfindung des Buchdrucks war sie eine schiere Notwendigkeit, das Wissen des Lehrenden in die Mitschriften der Lernenden zu übertragen. Zu meiner Zeit als Student wurde dies in manchen Lehrveranstaltungen durch Verzicht auf jedwede begleitenden Medien (Skript, Literaturhinweise etc.) aufrechterhalten. Aus rein lernpsychologischer Sicht ist die Vorlesung als reiner Frontalunterricht aber alles andere als eine ideale Lernform. Daher möchte ich den Raum reduzieren, den der reine Frontalunterricht zukünftig in meinen Lehrveranstaltungen einnimmt.

7.2 eLearning

Intrinsische Motivation in IT-Berufen entsteht nicht zuletzt dadurch, dass Feedback häufig unmittelbar ist. Ob ein Arbeitsschritt sinnvoll und erfolgreich war, erfährt der Software-Engineer häufig direkt, indem er seinen Code kompiliert und ausprobiert.

Diese Unmittelbarkeit möchte ich durch den Einsatz von eLearning auch in der Lehre erreichen. Hierbei verstehe ich aber unter eLearning nicht die reine Weiterführung des Frontalunterrichts mit anderen Mitteln, d.h. Video-Clips. Stattdessen versuche ich Werkzeuge einzusetzen, die Studierende aktiv einbinden. In der im Sommersemester von mir vertretungsweise durchgeführten Lehrveranstaltung "Praktische Informatik 2" werden ich Tools wie Repl.it einsetzen, über die Studierenden Programmieraufgaben lösen können und Feedback durch das Tools unmittelbar und natürlich vertraulich bekommen.

7.3 Didaktische Methoden

Ich habe mich zu didaktischen Methoden informiert und an Workshops teilgenommen (s. Anhang). Das Konzept des Flipped Classrooms werde ich voraussichtliche in der Lehrveranstaltung Software-Architecture for Large Systems (Arbeitstitel), s. Abschnitt 3.3.3, einsetzen.

7.4 Wissensbeschaffung lernen

Ich möchte die Studierenden in Zukunft vermehrt zur selbständigen Wissensbeschaffung anregen. Das Umfeld der IT ist so hochdynamisch, dass das Wissen zu konkreten Methoden und Werkzeugen schnell veraltet. Wer nach dem Hochschulabschluss in der IT nicht kontinuierlich und selbständig hinzulernt, wird keine beruflichen Erfolg haben.

Ich werde daher versuchen, meine Rolle zukünftig weg von der Rolle eines Vortragenden oder Instruktors hin zu der Rolle eines Beraters für Studierende zu verlagern.

Anhang

Im Folgenden finden Sie meine bisherigen EvaSys-Evaluationsergebnisse sowie Teilnahmebescheinigungen besuchter Didaktik-Workshops.



HS Düsseldorf
Stabsstelle Strategie und Innovation
Evaluationsbüro

Münsterstraße 156
Gebäude 02, Raum 3.028
40476 Düsseldorf

Tel.: +49 211 4351 8014
E-Mail:
a.schulz-kleyenstueber@hs-duesseldorf.de

Prof. René Wörzberger (PERSÖNLICH)

Auswertungsbericht Lehrveranstaltungsevaluation an die Lehrenden

Sehr geehrte/r Frau/Herr Prof. Wörzberger,

Sie erhalten hier die Ergebnisse der Auswertung der Lehrveranstaltungsevaluation zur Veranstaltung
Software Engineering

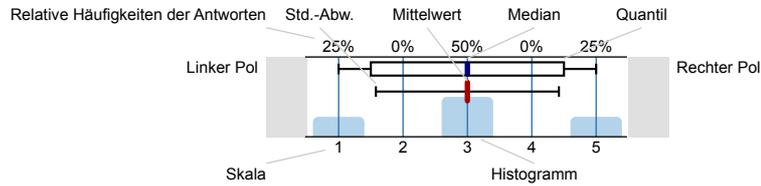
Fragebogen: fb05neu2

Mit freundlichen Grüßen

Auswertungsteil der geschlossenen Fragen

Legende

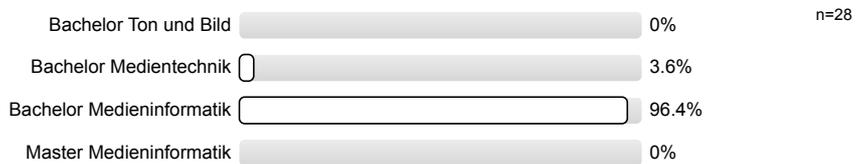
Fragestext



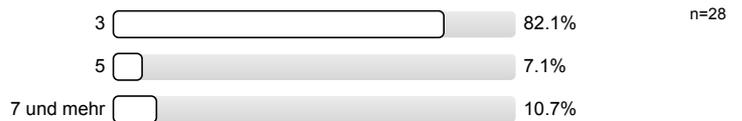
n=Anzahl
mw=Mittelwert
md=Median
s=Std.-Abw.
E.=Enthaltung

1. Studiengang / Fachsemester

1.1) In welchem Studiengang sind Sie eingeschrieben?

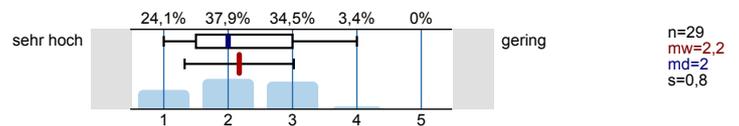


1.2) In welchem Fachsemester studieren Sie?

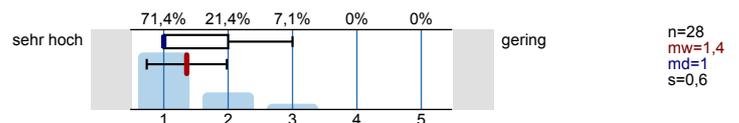


2. Allgemeines

2.1) Wie hoch ist Ihr Interesse an der Thematik der Lehrveranstaltung?



2.2) Wie hoch ist der Anteil der von Ihnen wahrgenommenen Sitzungen dieser Lehrveranstaltung?



2.3) Ergänzt bzw. erweitert die Thematik Ihre bisherigen Kenntnisse und Fähigkeiten?

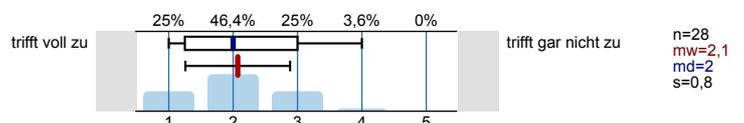


2.4) Würden Sie ein Tutorium für diese Lehrveranstaltung als sinnvoll erachten?

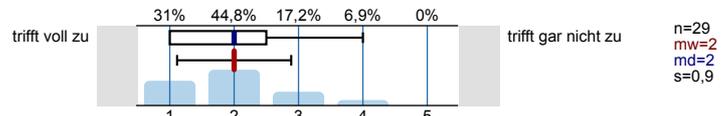


3. Inhalt

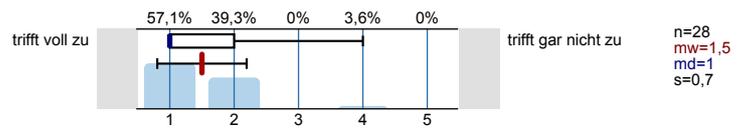
3.1) Die Struktur und Konzeption des Faches ist erkennbar und gut.



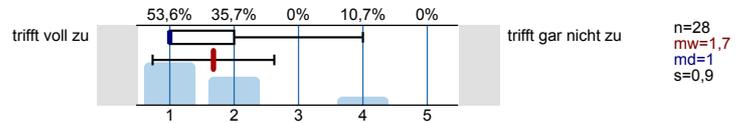
3.2) Bedeutung und Nutzen der behandelten Themen werden vermittelt.



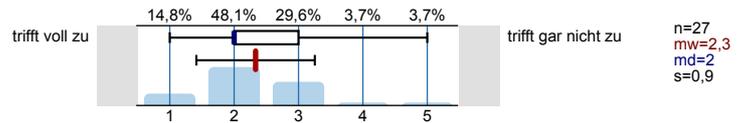
3.3) Auf neuere Entwicklungen und Erkenntnisse wird eingegangen.



3.4) Es wurden konkrete Beispiele gezeigt



3.5) Es wurden Zusammenhänge erläutert



3.6) Inhalte der Lehrveranstaltung wurden bereits in anderen Veranstaltungen des Studiengangs behandelt



4. Anforderungen/Betreuung

4.1) Ist das Lerntempo angemessen?



4.2) Gibt es nützliche Hinweise zum Selbststudium (Skript, Literatur etc.)?



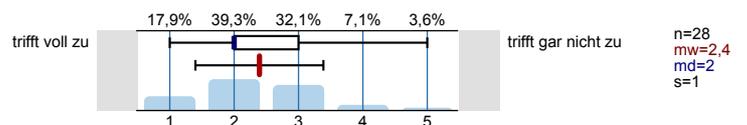
4.4) Verfügen Sie über das im Modulhandbuch ausgewiesene Vorwissen?



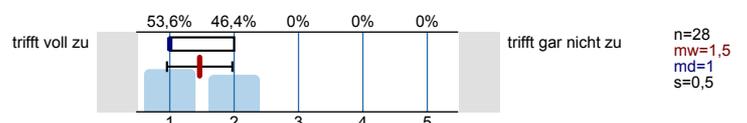
4.5) Wie hoch ist Ihr durchschnittlicher wöchentlicher Zeitaufwand zur Vor- und Nachbearbeitung dieser Veranstaltung? (in Stunden pro Woche, ohne separate Klausurvorbereitung):



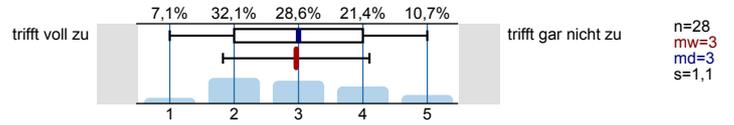
4.6) Der Stoff wird anschaulich und verständlich dargestellt



4.7) Der/die Lehrende lässt Fragen zu und beantwortet diese sachgerecht

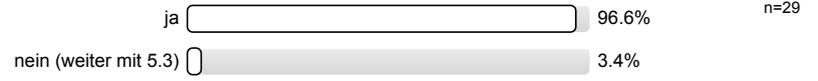


4.8) Die Prüfungsrelevanz der Inhalte ist klar geworden

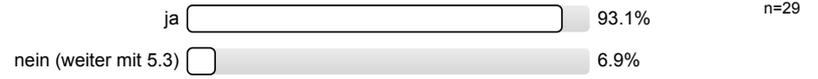


5. Rahmenbedingungen/Organisation

5.1) Entspricht die räumliche Planung der Lehrveranstaltung Ihren Vorstellungen?

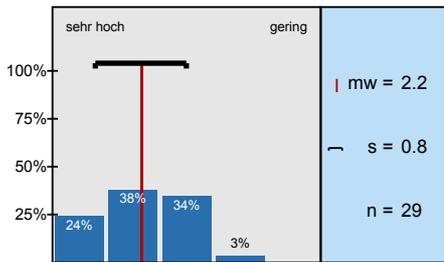


5.2) Entspricht die zeitliche Koordination der Lehrveranstaltung Ihren Vorstellungen?

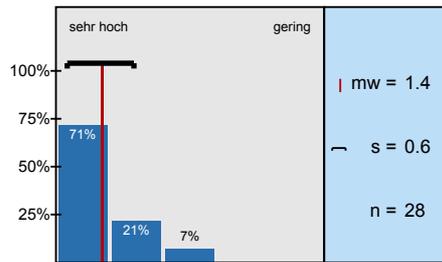


Histogramme zu den Skalafragen

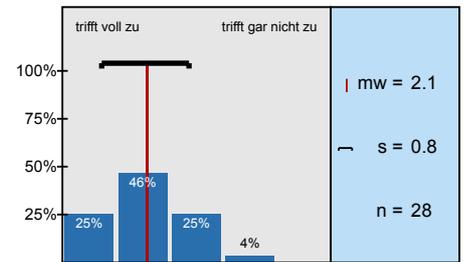
Wie hoch ist Ihr Interesse an der Thematik der Lehrveranstaltung?



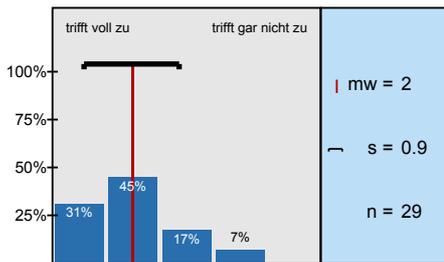
Wie hoch ist der Anteil der von Ihnen wahrgenommenen Sitzungen dieser



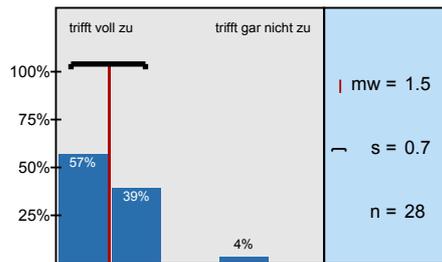
Die Struktur und Konzeption des Faches ist erkennbar und gut.



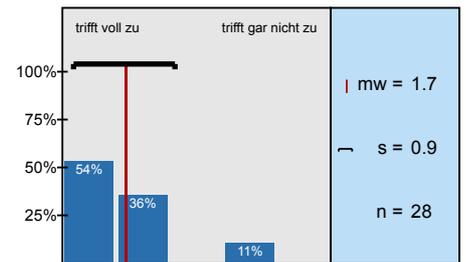
Bedeutung und Nutzen der behandelten Themen werden vermittelt.



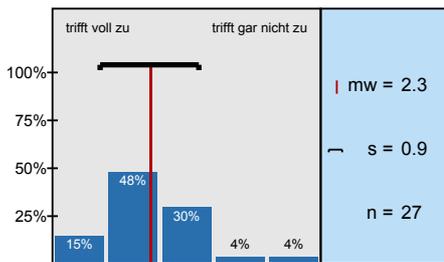
Auf neuere Entwicklungen und Erkenntnisse wird eingegangen.



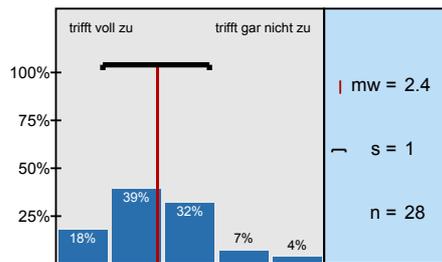
Es wurden konkrete Beispiele gezeigt



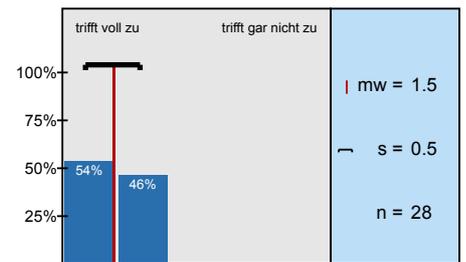
Es wurden Zusammenhänge erläutert



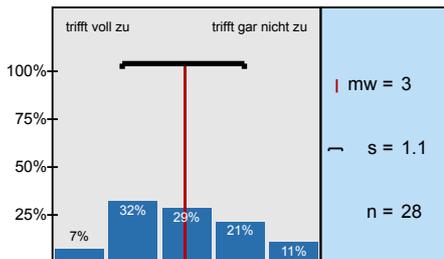
Der Stoff wird anschaulich und verständlich dargestellt



Der/die Lehrende lässt Fragen zu und beantwortet diese sachgerecht



Die Prüfungsrelevanz der Inhalte ist klar geworden



Profillinie

Teilbereich: Fachbereich Medien
 Name der/des Lehrenden: Prof. René Würzberger
 Titel der Lehrveranstaltung: Software Engineering
 (Name der Umfrage)

Verwendete Werte in der Profillinie: Mittelwert

2. Allgemeines

2.1) Wie hoch ist Ihr Interesse an der Thematik der Lehrveranstaltung?	sehr hoch		gering	n=29	mw=2,2	md=2,0	s=0,8
2.2) Wie hoch ist der Anteil der von Ihnen wahrgenommenen Sitzungen dieser Lehrveranstaltung?	sehr hoch		gering	n=28	mw=1,4	md=1,0	s=0,6

3. Inhalt

3.1) Die Struktur und Konzeption des Faches ist erkennbar und gut.	trifft voll zu		trifft gar nicht zu	n=28	mw=2,1	md=2,0	s=0,8
3.2) Bedeutung und Nutzen der behandelten Themen werden vermittelt.	trifft voll zu		trifft gar nicht zu	n=29	mw=2,0	md=2,0	s=0,9
3.3) Auf neuere Entwicklungen und Erkenntnisse wird eingegangen.	trifft voll zu		trifft gar nicht zu	n=28	mw=1,5	md=1,0	s=0,7
3.4) Es wurden konkrete Beispiele gezeigt	trifft voll zu		trifft gar nicht zu	n=28	mw=1,7	md=1,0	s=0,9
3.5) Es wurden Zusammenhänge erläutert	trifft voll zu		trifft gar nicht zu	n=27	mw=2,3	md=2,0	s=0,9

4. Anforderungen/Betreuung

4.6) Der Stoff wird anschaulich und verständlich dargestellt	trifft voll zu		trifft gar nicht zu	n=28	mw=2,4	md=2,0	s=1,0
4.7) Der/die Lehrende lässt Fragen zu und beantwortet diese sachgerecht	trifft voll zu		trifft gar nicht zu	n=28	mw=1,5	md=1,0	s=0,5
4.8) Die Prüfungsrelevanz der Inhalte ist klar geworden	trifft voll zu		trifft gar nicht zu	n=28	mw=3,0	md=3,0	s=1,1

Auswertungsteil der offenen Fragen

3. Inhalt

3.7) In welcher Veranstaltung wurden bereits gleiche Inhalte behandelt und um welche Inhalte handelt es sich?

OOP

Webeng

OOP1+2, DBS1+2, WebAb

Inhalte aus dem UML-Kapitel zu einem kleinen Teil schon in Datenbanken II

UML in Datenbanken, aber nicht so ausführlich wie in SWE

es überschneidet sich teilweise mit OOP (Threads), manche Programmieraufgaben

OOP1: Grundlagen von Programmiersprachen

OOP

HTIP → Web Engineering

Build tools → Web Frameworks

JS → Web Apps

Viele Themen werden erst im nächsten Semester behandelt AB: Webeng
viel wird erwartet was noch nicht Thema war ...

4. Anforderungen/Betreuung

4.3) Welche Hinweise zum Selbststudium würden Sie sich wünschen?

Mehr Sources als die offizielle Seite eines Tools, wir sitzen im Praktikum 80% der Zeit daran, das Tool zu verstehen, statt die Aufgaben im ~~Praktikum~~ ^{Praktikum} zu ~~erklären~~ ^{erklären}.

5. Rahmenbedingungen/Organisation

5.3) Welche Probleme gibt es bei der räumlichen und zeitlichen Planung?

Raum: Steckdosen funktionieren nicht, Sitze nicht gepolstert

Laut PO sollte es nur ein Seminar geben. Hier ist jedoch wöchentlich ein Praktikum inkl Testaten. Dadurch entsteht viel mehr Zeitaufwand als in den letzten Semestern... komplizierteste Praktikumaufg.

6. Fazit

6.1) Was hat Ihnen besonder gut und was hat Ihnen weniger gut an der Lehrveranstaltung gefallen?

Erklärung von Tools

Relevante Themen

↳ Sinnvoll und Nutzbear

+ Praxisbezug

- Praktika aufwendig, Vorlesung wenig hilfreich

- + Wissen aus Industrie
- Unzahl an Tools, alle nur angeschnitten
- + ars Nova
- Klausuraufbau
- viele Beispiele

Schön strukturiert, gute Beispiele.

Hier lernt ~~man~~ man all das, was man in einem "echten" Job wissen sollte!

Besonders gut: Themen werden sehr ausführlich erklärt und dadurch gut verständlich
Auf Fragen wird eingegangen

- sehr gute praktische Beispiele während der Vorlesung
- gute Erklärungen und übersichtliche Folien
- Praktikas ergänzen das Wissen aus der Vorlesung und prüfen dieses nicht nur ab

gut: - Beispiele (Code + Tools)

nicht so gut: - Praktika werden einen oft ins kalte Wasser, Tools noch unbekannt oder kaum bekannt.

- durchgehendes Praxisbeispiel ShipMe
 - Live Coding / Vorführung von Beispielen
 - Das Umfrages tool
 - Aber man lernt viele Tools im Praktikum kennen, die in der Praxis relevant sind
 - man darf frei entscheiden, mit welcher IDE man arbeitet
 - Benutzung der Kommandozeile
-
- viele verschiedene Tools, die alle nur grob angerissen werden*
 - Zusammenhang zwischen Vorlesung und Praktikum ist nicht immer ganz klar

besonders gut:

- interaktive Vorlesung
- "live coding"
- viele verständliche Beispiele

weniger gut:

- im Praktikum viel vorgegeben
- viel Zeitaufwand

Der Professor ist sehr nett und verständnisvoll. Man versteht ihn sehr gut.

Besonders gut: viele praktische Beispiele

weniger gut: Zusammenhänge werden sehr spät oder teilweise gar nicht dargestellt (z.B. Maven)

→ im Praktikum werden daher nur die Schritte befolgt (ohne das zu hinterfragen)

! Detailreichtum und Codebeispiele.
Offenes und Hilfsbereites Verhalten.



⊕ - Tempo
- Fragen beantworten

⊖ - Fehlende Relation von
Vorlesung zu Praktikum

Engagement

- ⊖ Praktikum zu viel Vorwissen und erwartet
dadurch sehr viel Zeitaufwand dafür ...
in den letzten Semestern & bei SWE ganz anders
Jahren
- ⊕ Vorlesungen + Skript

6.2) Haben Sie Verbesserungsvorschläge? Wenn ja, welche?

nicht immer das selbe Programm (Address book)
im Praktikum verwenden.

Viel Aufwand für das Praktikum
↳ Sollte ein wenig gemindert werden!

mehr Praktikumsbezug in der Vorlesung wünschenswert

+ Problemlösung

bessere Erklärung der Hintergründe
der einzelnen Tools, etc

Schwer verständlich, wenn man kein Vorwissen
hat

Tool mit Beispiel vorstellen, dann Praktikum zu Tool

- manchmal wird man im Praktikum von der Fülle der Tools erschlagen
[Inkellij statt Eclipse, aber das muss jeder selber entscheiden]

~~etc~~

* zumindest einige davon etwas mehr vertiefen, Nutzen zeigen

Mehr auf die im Praktikum behandelten Themen
eingehen.

Zich würde mir mehr Beispiele, gerade für das Praktikum bezogen wünschen. Die Fragen sind manchmal etwas schwer zu verstehen.

Praktikum und Vorlesung sind Welten voneinander entfernt, was ihren jeweiligen Anspruch angeht.

Mehr Bezug auf die Praktika.

Auf Praktikumsinhalte in den Vorlesungen eingehen, ~~vor~~ bevor das Praktikum stattfindet, sonst erarbeitet man sich alles selber und man kann sich die Vorlesung sparen.

Dinge wie beispielsweise HTTP entweder weglassen, oder vernünftig erklären.

Im Praktikum nicht der VL vorgreifen!

wieder Seminar mit Vorträgen (z.B.), keine Praktikas

TH Köln · Gustav-Heinemann-Ufer 54 · 50968 Köln

An
die Lehrenden
der TH Köln

Im Hause

Köln, 17.12.2018

Ergebnisse Lehrveranstaltungsbeurteilung

Sehr geehrte Damen und Herren,

im Anhang finden Sie die Ergebnisse Ihrer Lehrveranstaltungsbeurteilung aus dem aktuellen Semester.

Die Bewertungsbögen zielen auf eine Selbsteinschätzung der Studierenden zu unterschiedlichen Kompetenzbereichen, die jeweils in Frageblöcken abgebildet werden. Die im Einzelnen genannten Anforderungen und Lernziele treffen aber nicht notwendigerweise alle auch auf Ihre Lehrveranstaltung zu. Maßgeblich sind in diesem Zusammenhang die in den Modulbeschreibungen niedergelegten kompetenzorientierten Lernziele.

Zusammenfassende Indikatoren-Wertungen werden nur für solche Fragegruppen ausgewiesen, die für alle Lehrveranstaltungen trotz unterschiedlicher Anforderungen und Lernziele zutreffen.

Neben der grafischen Darstellung aller Einzelergebnisse gibt es einen Überblick über die Bewertungsergebnisse in Form sogenannter Profillinien. Dabei werden die jeweils einer Fragegruppe zugehörigen Frageaspekte basierend auf dem arithmetischen Mittelwert im Zusammenhang dargestellt.

Von den Mittelwertangaben (mw) zur Workload muss für den korrekten Wert systembedingt eine Stunde abgezogen werden. Am Ende der Ergebnisübersicht finden Sie ggf. eingescannte handschriftliche Anmerkungen.

Darüber hinaus steht Ihnen das Kompetenzteam Hochschuldidaktik zur Verfügung, um mit Ihnen Evaluationsergebnisse zu interpretieren, Feedbacks Ihrer Studierenden zu reflektieren oder Lehrideen zu vertiefen sowie anderes mehr. Sie erreichen das Team per Mail (hochschuldidaktik@th-koeln.de) oder telefonisch (unter 8275 – 3820).

Mit freundlichen Grüßen

Im Auftrag

Michael Rüdel

**Technology
Arts Sciences
TH Köln**

Dr. Michael Rüdel
Leiter Hochschulreferat
Qualitätsmanagement
+49 221-8275-3965
michael.ruedel@th-koeln.de
B4.270
Claudiusstraße 1
50678 Köln

Technische Hochschule Köln

Postanschrift:
Gustav-Heinemann-Ufer 54
50968 Köln

Sitz des Präsidiums:
Claudiusstraße 1
50678 Köln

www.th-koeln.de

Steuer-Nr.: 214/5805/0184
USt-IdNr.: DE 122653679

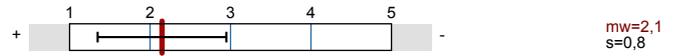
Bankverbindung:
Sparkasse KölnBonn
IBAN DE34 3705 0198 1900 7098 56
BIC COLSDE33

Lehrveranstaltungsbewertungen WS 2018-19

F07_201825111_W18 ()
Erfasste Fragebögen = 19

Globalwerte

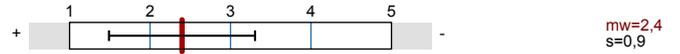
Rahmenbedingungen der Lehrveranstaltung



Die/Der Lehrende



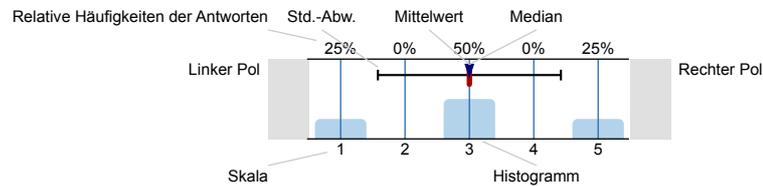
Studieninteresse / Selbstkompetenz



Auswertungsteil der geschlossenen Fragen

Legende

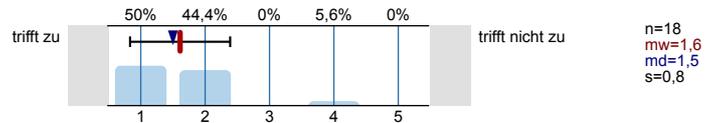
Fragestext



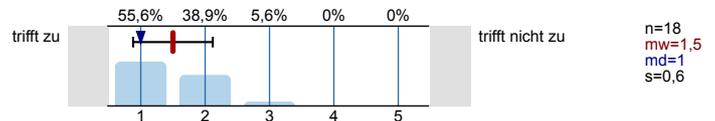
n=Anzahl
mw=Mittelwert
md=Median
s=Std.-Abw.
E.=Enthaltung

Rahmenbedingungen der Lehrveranstaltung

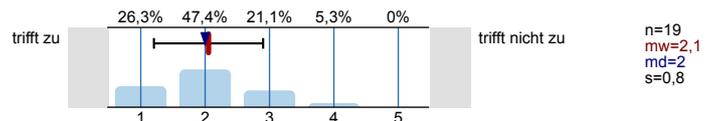
Die Lernziele dieser Lehrveranstaltung wurden zu Beginn deutlich gemacht



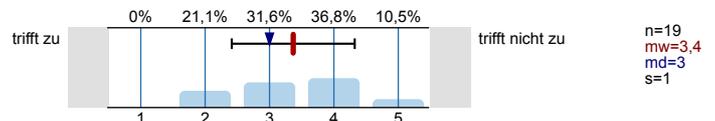
Die Bedeutung dieser Lehrveranstaltung für meinen Studiengang ist für mich deutlich



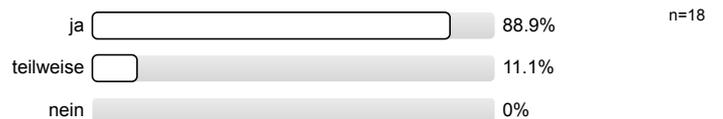
Die Inhalte der einzelnen Veranstaltungen sind klar gegliedert und nachvollziehbar



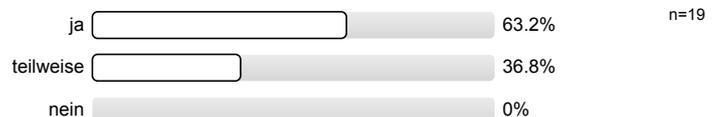
Die Prüfungsanforderungen sind eindeutig benannt worden



Es sind Hilfsmittel (Skripte, Folien, E-Learning...) verfügbar, die meinen Lernprozess unterstützen

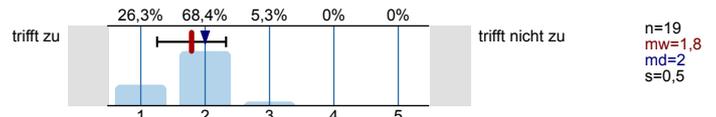


Ich habe die fachlichen Voraussetzungen, um d. Lehr-veranst.g gut folgen bzw. mich aktiv beteiligen zu können

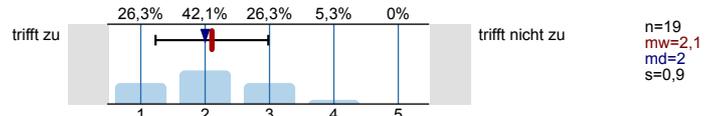


Die/Der Lehrende

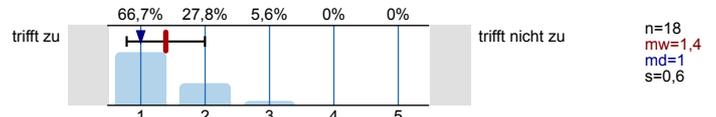
ist nach meinem Eindruck sehr am Lernerfolg der Studierenden interessiert



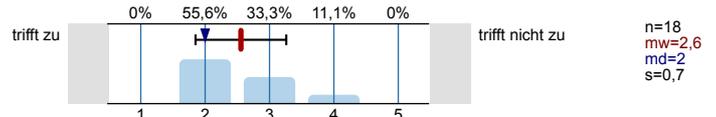
unterstützt meinen Lernprozess durch die didaktische Gestaltung der Lehrveranstaltung



ist nach meinem Eindruck offen für Fragen und Anregungen



ermöglicht mir die Einschätzung meines Lernerfolgs auch unabhängig von Prüfungen

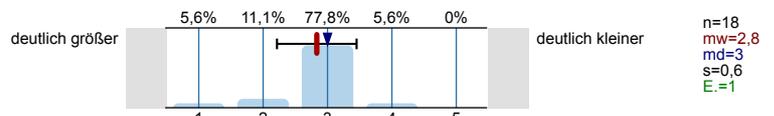


Workload

Wie viel Zeit wenden Sie durchschnittlich pro Woche für die Vor-/Nachbereitung dieser Lehrveranstaltung (V und Ü) auf?



Der zeitliche Aufwand ist gegenüber Veranstaltungen mit gleicher Anzahl Kreditpunkte (CP)

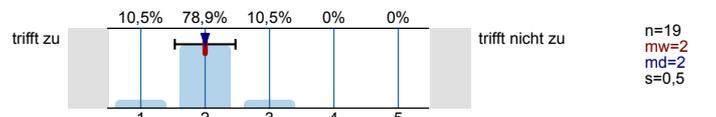


Den zeitlichen Aufwand für diese Lehrveranstaltung finde ich insgesamt angemessen

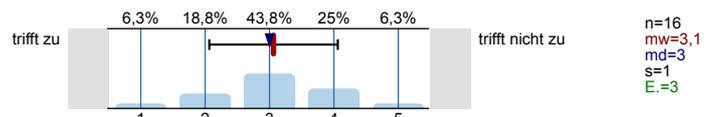


Fach- und Methodenkompetenz - Bitte beantworten Sie alle Fragen nur für diese Lehrveranstaltung

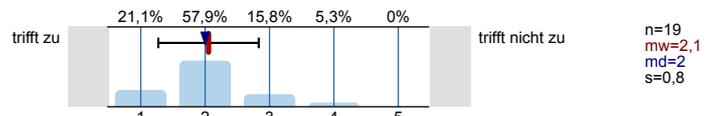
Ich habe bislang in dieser Lehrveranstaltung viel gelernt



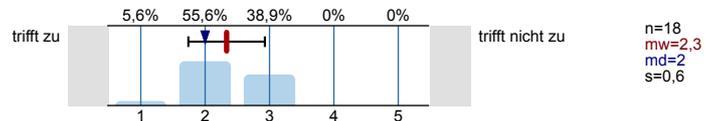
Ich habe im Rahmen d. Lehrveranstaltung meine Arbeitstechniken (Recherche, Dokumentation von Ergebnissen ...) verbessert



Ich kann wichtige Begriffe und Sachverhalte zu den behandelten Themen wiedergeben



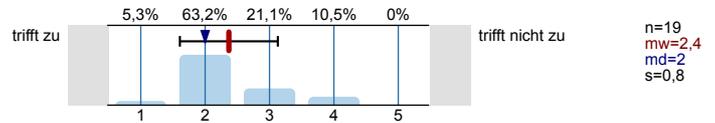
Ich kann die bisherigen Inhalte d. Lehrver-anstaltung zusammenfassen u. auch schwie-rige Sachverhalte anschaulich darstellen



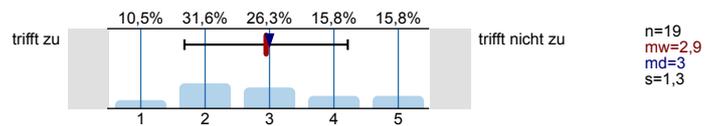
Ich kann Inhalte und Methoden des Fachs auf praktische Fragestellungen anwenden



Ich kann Fragestellungen des Fachs selbständig analysieren und bewerten



Ich habe eigene Ideen, wie bekannte Sachverhalte weiterentwickelt werden können



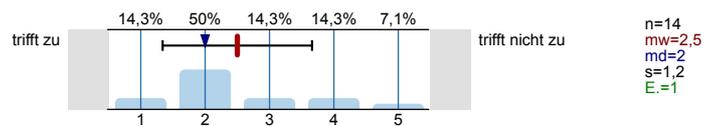
Praktikum

Haben Sie am Praktikum teilgenommen?



Falls ja:

Meine theoretischen Kenntnisse reichen für die Durchführung der Praktikumsversuche aus



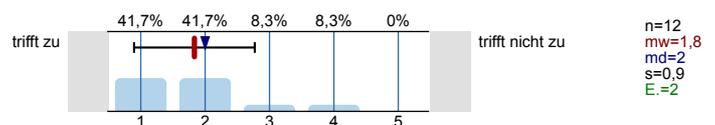
Durch die Versuche bin ich in der Lage, verschiedene Arbeits- und Messmethoden selbständig anzuwenden



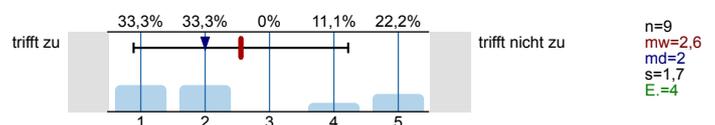
Ich erkenne zwischen den Versuchen einen klaren, nachvollziehbaren Zusammenhang



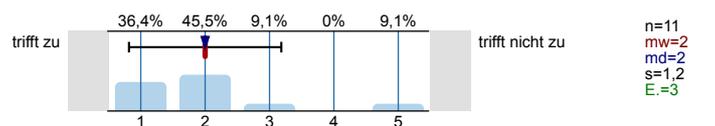
Ich komme mit der vorgegebenen Zeit aus, Versuche durchzuführen u. zu protokollieren



Die Rückmeldungen zu d. Versuchsproto-kollen sind verständlich u. nachvollziehbar



Ich werde im Praktikum gut betreut



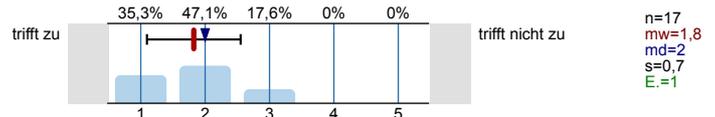
Übung

Haben Sie an der Übung zur Vorlesung teilgenommen?

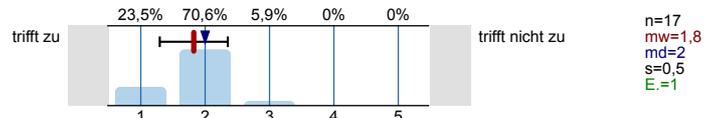


Falls ja:

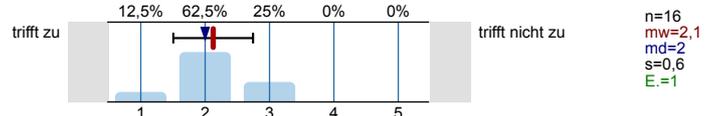
Durch die Übung kann ich Vorlesungsinhalte jetzt besser verstehen



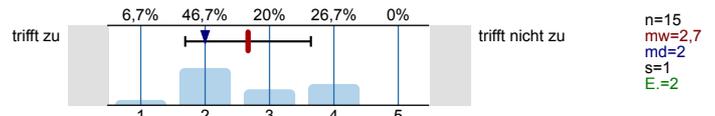
Die Übung hilft mir nachzuvollziehen, wie Theorie und Praxis miteinander verknüpft sind



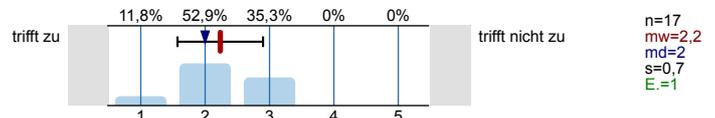
Durch die Übung erreiche ich ein besseres Verständnis, wie die einzelnen Themengebiete zusammenhängen



Die Rückmeldungen zu d. Übungsergebnissen sind verständlich u. nachvollziehbar

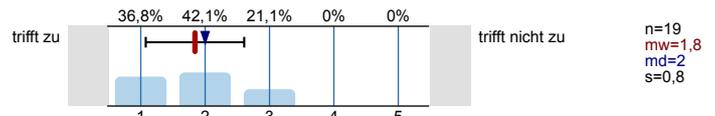


Ich werde in der Übung gut betreut

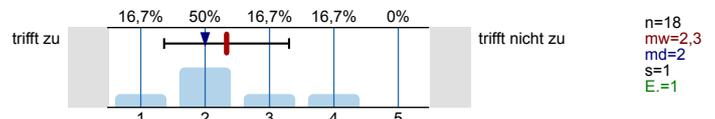


Studieninteresse / Selbstkompetenz

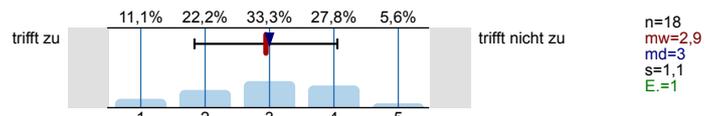
Die Lehrveranstaltung hat mein Interesse am Thema geweckt bzw. gefestigt



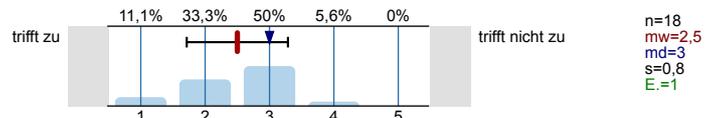
Ich setze mich regelmäßig mit dem Thema der Lehrveranstaltung auseinander



Ich habe Ideen, wie ich das Thema selbständig weiter vertiefen will

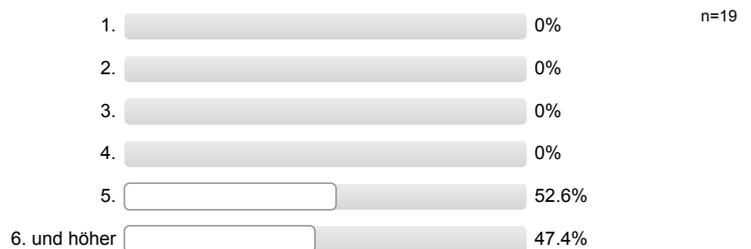


Aus dieser Lehrveranstaltung nehme ich Motivation für mein weiteres Studium mit



Strukturdaten

In welchem Fachsemester studieren Sie?



Ihr Geschlecht?



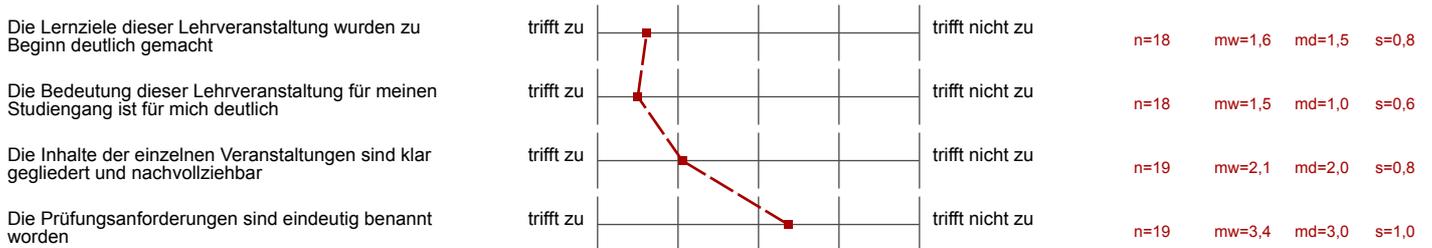
Vielen Dank für Ihre Rückmeldung!

Profillinie

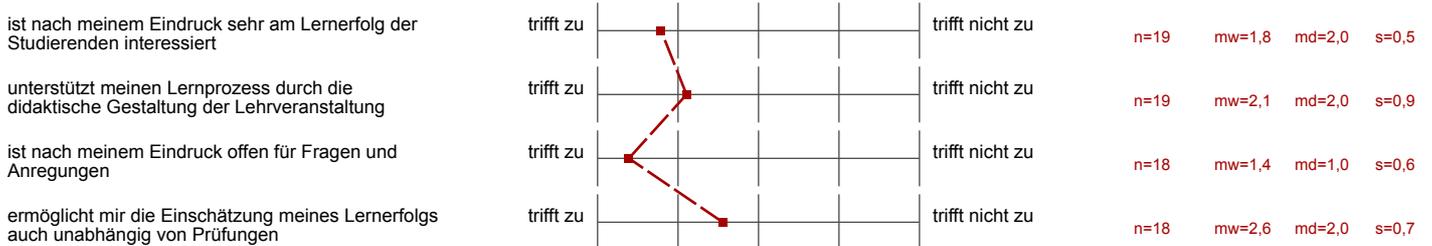
Teilbereich: Fakultät 07
 Name der/des Lehrenden: Lehrveranstaltungsbewertungen WS 2018-19
 Titel der Lehrveranstaltung: F07_201825111_W18
 (Name der Umfrage)

Verwendete Werte in der Profillinie: Mittelwert

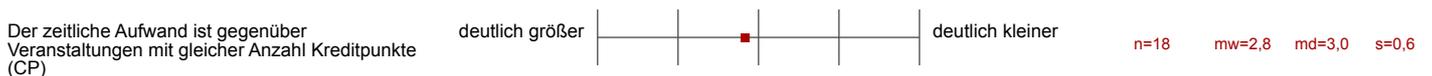
Rahmenbedingungen der Lehrveranstaltung



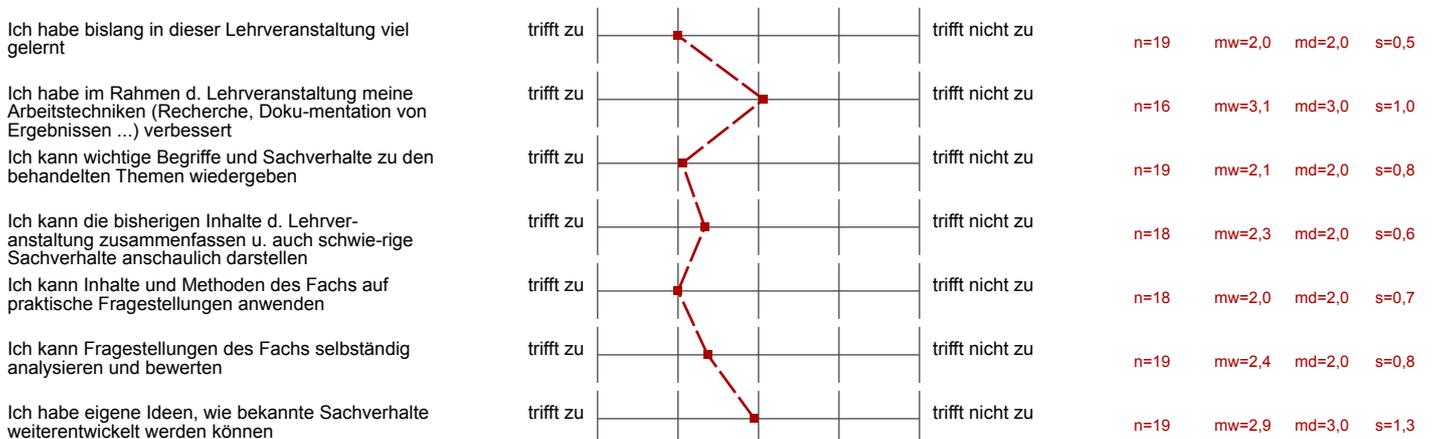
Die/Der Lehrende



Workload



Fach- und Methodenkompetenz - Bitte beantworten Sie alle Fragen nur für diese Lehrveranstaltung



Praktikum

Meine theoretischen Kenntnisse reichen für die Durchführung der Praktikumsversuche aus	trifft zu					trifft nicht zu	n=14	mw=2,5	md=2,0	s=1,2
Durch die Versuche bin ich in der Lage, verschiedene Arbeits- und Messmethoden selbständig anzuwenden	trifft zu					trifft nicht zu	n=14	mw=1,9	md=2,0	s=0,6
Ich erkenne zwischen den Versuchen einen klaren, nachvollziehbaren Zusammenhang	trifft zu					trifft nicht zu	n=11	mw=1,5	md=2,0	s=0,5
Ich komme mit der vorgegebenen Zeit aus, Versuche durchzuführen u. zu protokollieren	trifft zu					trifft nicht zu	n=12	mw=1,8	md=2,0	s=0,9
Die Rückmeldungen zu d. Versuchsprotokollen sind verständlich u. nachvollziehbar	trifft zu					trifft nicht zu	n=9	mw=2,6	md=2,0	s=1,7
Ich werde im Praktikum gut betreut	trifft zu					trifft nicht zu	n=11	mw=2,0	md=2,0	s=1,2

Übung

Durch die Übung kann ich Vorlesungsinhalte jetzt besser verstehen	trifft zu					trifft nicht zu	n=17	mw=1,8	md=2,0	s=0,7
Die Übung hilft mir nachzuvollziehen, wie Theorie und Praxis miteinander verknüpft sind	trifft zu					trifft nicht zu	n=17	mw=1,8	md=2,0	s=0,5
Durch die Übung erreiche ich ein besseres Verständnis, wie die einzelnen Themengebiete zusammenhängen	trifft zu					trifft nicht zu	n=16	mw=2,1	md=2,0	s=0,6
Die Rückmeldungen zu d. Übungsergebnissen sind verständlich u. nachvollziehbar	trifft zu					trifft nicht zu	n=15	mw=2,7	md=2,0	s=1,0
Ich werde in der Übung gut betreut	trifft zu					trifft nicht zu	n=17	mw=2,2	md=2,0	s=0,7

Studieninteresse / Selbstkompetenz

Die Lehrveranstaltung hat mein Interesse am Thema geweckt bzw. gefestigt	trifft zu					trifft nicht zu	n=19	mw=1,8	md=2,0	s=0,8
Ich setze mich regelmäßig mit dem Thema der Lehrveranstaltung auseinander	trifft zu					trifft nicht zu	n=18	mw=2,3	md=2,0	s=1,0
Ich habe Ideen, wie ich das Thema selbständig weiter vertiefen will	trifft zu					trifft nicht zu	n=18	mw=2,9	md=3,0	s=1,1
Aus dieser Lehrveranstaltung nehme ich Motivation für mein weiteres Studium mit	trifft zu					trifft nicht zu	n=18	mw=2,5	md=3,0	s=0,8

Auswertungsteil der offenen Fragen

Strukturdaten

Anmerkungen? (Handschriftliche Anmerkungen gehen direkt an die Lehrenden, daher keine Gewährleistung der Anonymität!)

- vortragen ohne Begeisterung bzw. etwas lustlos
- zu oft "ehm"

Übungen unübersichtlich, funktionieren oftmals nicht.

Fürs Praktikum fehlt teilweise theoretisches Verständnis

Vorlesungen und Übungen ~~sehr~~ eher schwach/schwammig
 Folien helfen nicht bei der Praktikumsvorbereitung
 Praktikumsvorbereitungen meistens mit viel geögeln möglich, aufwendig
 Folien ~~enthalten~~ enthalten viele Fehler, ~~...~~

- multi OS (an verschiedene Line Endings denken)
 - PDFs (Code Bsp. sind nicht kopierbar, Zeilen unbrüche)
 - Sie haben laut Stundenplan viel mehr Zeit, nutzen Sie diese, wenn es Zeitprobleme gibt.
 - Sehr wichtigen Fach. Ggf. zum Pflichtfach für Informatiker machen? Es ist sonst schade um unser Wissen.
- Übungen kommen meist zu kurz
 Danke für Ihr Engagement!!

Das 2. Praktikum hat nicht so funktioniert wie von ihnen vorgesehen..

Kurze Überreden zur Bedienung von git, waren etc wären sicher Grad ~~...~~ von Studenten willkommen, auch wenn diese selbst zu erstellen eine gute Übung ist.

Teilnahmebescheinigung

**Technology
Arts Sciences
TH Köln**

Prof. Dr. René Würzberger

hat am 04. Juni 2018 an dem folgenden Workshop teilgenommen:

Das Flipped Classroom-Modell praktisch gestalten

Lernziele

Die Teilnehmer*innen können geeignete Lernziele für die Selbstlernphase und die Präsenzphase definieren und verschiedene Möglichkeiten zur Aufbereitung von Selbstlernmaterial benennen und die für ihre Lehrveranstaltung adäquaten auswählen. Darüber hinaus sind die Teilnehmer*innen in der Lage, motivationsrelevante Rahmenbedingungen für die Studierenden zu analysieren und konstruktiv zu gestalten und können eine Auswahl an möglichen Methoden für die Präsenzphase benennen und einsetzen.

Arbeitsweise

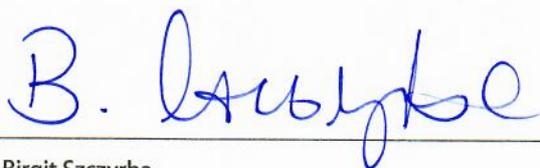
- Input/Gruppenarbeit
- offene Werkstatt/Diskussion
- kollegialer Austausch

Referent

- Daniel Al Kabbani (Creaversity)

Umfang

- 8 AE Workshop



Dr. Birgit Szczyrba

Zentrum für Lehrentwicklung

Leiterin Team Hochschuldidaktik

Köln, 07.06.2018

Dr. Birgit Szczyrba

Zentrum für Lehrentwicklung
Leiterin Team Hochschuldidaktik

T: +49 221-8275-3622

E: birgit.szczyrba@th-koeln.de

Campus Südstadt

Claudiusstraße 1

50678 Köln

Raum E 1.03

Teilnahmebescheinigung

Teilnehmer/in: Herr Dr. René Wörzberger, Hochschule Düsseldorf
Veranstaltungstitel: „Stimmtraining: Stimme und Vorlesungsrhetorik“
Datum: 12./13. und 14.09.2017
Veranstaltungsort: Hochschule Bonn-Rhein-Sieg
Dauer/Umfang: 10.00 – 17.00 Uhr (jeweils 8 LE pro Tag)
Zuordnung: Themenbereich Schlüsselqualifikationen für Lehrende



Workshopschwerpunkte:

- Entspannung und Körperstimmtraining
- Impulse und Körpersprache – der lebendige Vortrag
- die Kriterien der Hörverständlichkeit
- die Redeplanung
- Feedback der stimmlichen und rhetorischen Präsentation
- Vortrag/Vorlesung in die Köpfe der Anderen bringen
- Wie kann ich interessieren und überzeugen?

Methoden/Arbeitsformen: Impulsvortrag, Gruppen-, Partner- und Einzelarbeit, Diskussion

St. Augustin, 14.09.2017


Dipl. Sprechwiss. Günter Wirth
Referent


Prof. Dr. Tobina Brinker
hdw nrw

Netzwerk Hochschuldidaktische Weiterbildung Nordrhein-Westfalen hdw nrw**Teilnahmebescheinigung**

Teilnehmer/in: Herr Prof. Dr. René Würzberger, Hochschule Düsseldorf
Veranstaltungstitel: „Digitale Lehre - ILIAS Online Kommunikation mit Forum, Chat und Wiki“
Datum: 07.05.2018
Veranstaltungsort: Webinar Online
Dauer/Umfang: 09.30 – 12.00 Uhr (4 LE)
Zuordnung: Themenbereich DL

**Workshopschwerpunkte:**

- Was sind Chat, Forum und Wiki in ILIAS
- Einsatzszenarien aus der Hochschulpraxis
- Hinweise zur Integration der Tools in die eigene Lehre

Methoden/Arbeitsformen: Impulsvortrag, Gruppen-, Partner- und Einzelarbeit, Diskussion

Anmerkungen:

Der Workshop wird für das Hochschuldidaktische Zertifikat I und II im Umfang von 4 LE anerkannt. Das Programm des Netzwerkes hdw nrw ist akkreditiert durch die AKKO (Akkreditierungskommission der Deutschen Gesellschaft für Hochschuldidaktik).

Bielefeld, 07.05.2018

i. A. Skonbor
Dipl.-Päd. André Mersch
Referent

i. A. Skonbor
Prof. Dr. Tobina Brinker
hdw nrw

Teilnahmebescheinigung

Teilnehmer/in: Herr Prof. Dr. René Wörzberger, Hochschule Düsseldorf
Veranstaltungstitel: „Methodische Gestaltung der eigenen Lehre“
Datum: 28.05.2018
Veranstaltungsort: Hochschule Bonn-Rhein-Sieg
Dauer/Umfang: 10.00 – 17.00 Uhr (8 LE)
Zuordnung: Basiskurse (BK)



Workshopschwerpunkte:

- Veranstaltungsplanung: Rahmenbedingungen und Lernvoraussetzungen analysieren, Lehrziele setzen, Methoden auswählen, Inhalte reduzieren
- Veranstaltungsformen: Vorlesung, Übung, Seminar, Labor, Exkursion u.v.m.
- Methoden: Methodenwahl, Methodenvielfalt erleben
- Kleine Helfer: Orientierung geben, Anfangssequenzen auflockern, Interaktion in der Vorlesung, Transfer vorbereiten

Methoden/Arbeitsformen: Impulsvortrag, Gruppen-, Partner- und Einzelarbeit, Diskussion

Anmerkungen:

Der Workshop wird für das Hochschuldidaktische Zertifikat I und II im Umfang von 8 LE anerkannt. Das Programm des Netzwerkes hdw nrw ist akkreditiert durch die AKKO (Akkreditierungskommission der Deutschen Gesellschaft für Hochschuldidaktik).

Rheinbach, 28.05.2018

A handwritten signature in blue ink that reads 'Brall'.

Dr. Stefan Brall
Referent

A handwritten signature in blue ink that reads 'Brinker'.

Prof. Dr. Tobina Brinker
hdw nrw